MAUS Goes Iterative

Florian Schiel

Bavarian Archive for Speech Signals (BAS)
c/o Institut für Phonetik und Sprachliche Kommunikation

University of Munich, Schellingstr. 3, 80799 München, Germany
schiel@bas.uni-muenchen.de

Abstract

In this paper we describe further developments of the MAUS system and announce a free-ware software package that may be downloaded from the 'Bavarian Archive for Speech Signals' (BAS) web site. The quality of the MAUS output can be considerably improved by using an iterative technique. In this mode MAUS will calculated a first pass through all the target speech material using the standard speaker-independent acoustical models of the target language. Then the segmented and labelled speech data are used to re-estimated the acoustical models and the MAUS procedure is applied again to the speech data using these speaker-dependent models. The last two steps are repeated iteratively until the segmentation converges. The paper describes the general algorithm, the German benchmark for evaluating the method as well as some experiments on German target speakers.

1. Introduction

With increasing sizes of empirical language resources tools for a fully automatic annotation of speech become more and more important. The 'Munich AUtomatic Segmentation' (MAUS) is a well established technique to produce segmentations and labellings (S&L) on the phonetic/phonemic level in a quality that compares to that of inter-labeller agreements of human labellers (Kipp et al., 1996), (Schiel, 1999). Compared to the inter-labeller agreement of three trained phoneticians MAUS currently (2003) achieves a normalized performance of 96.99% on our combined test and development set.

However, this performance may decrease considerably under certain conditions of the chosen data set

- when the acoustical environment changes dramatically (office vs. telephone recording, background noise etc.)
- when the speaker shows a dialect not seen in the MAUS training set,
- when MAUS is to be used in a different language than German and only very few data sets are available for the training of the underlying Hidden Markov Models.

For example, in a recent study with Australian English we found that the MAUS performance is not acceptable when using models trained on the American part of the Verbmobil corpus (Weilhammer et al., 2002). The re-training of the HMM to the data of five Australian speakers did not work either, probably because the inter-speaker variability between training and target speakers was too large.

Based on these experiences and with the aim to further improve the MAUS performance in general we propose a new iterative segmentation method `maus.iter` that may be used for the S&L of 'unknown material' which is not well represented in the acoustical models of MAUS.

In the following we will very briefly describe the traditional MAUS method, more in detail the new iterative method and some interesting results from experiments on our controlled benchmark system. We will then discuss the question how many data are necessary for the new method and finally give a short description about the currently available download package of the MAUS system at BAS.

2. Traditional MAUS

In a nutshell, traditional `maus` first computes a statistically weighted graph containing all likely pronunciation variants based on the orthographic and lexical representation of the utterance in question, and then aligns this model to the recorded speech signal using speaker- independent acoustical HMMs and the Viterbi algorithm (Young et al., 1995). Thus, the found S&L is the result of a combined optimization of acoustic likelihoods produced by the HMMs and the total probability along a path through the stochastic pronunciation graph. The key of the MAUS system is a set of stochastic re-write rules that can be learned automatically from a corpus of manually labelled reference data of the target language. In previous studies we found that about 1h of recorded and annotated speech for this reference material is sufficient to yield satisfactory results (Kipp, 1999). The acoustical models of `maus` are simple left-to-right HMM representing monophones and are trained on approx. 1h 40min of manually segmented speech from the 'Kiel Corpus of Spontaneous Speech'. Like (Kessens et al., 2001) we found that using HMM optimized for automatic speech recognition perform worse than models trained on a fixed segmentation – even if the amount of manually segmented data is much smaller (5%).

The drawback of the traditional `maus` tool is that speech signals recorded under different conditions than the reference material does not perform well enough. This has two reasons (possibly more):

- The set of stochastic re-write rules is not suitable for the new conditions.
  Of course this is the case when we switch to another dialect or even to another language. The problems involved with dialect/language change are not discussed

in this paper. Refer to (Beringer, 2002) or (Beringer, 2003) for possible solutions.

- The set of acoustical HMMs is not suitable for the new conditions.
  These may include the technical recording setup, the background noise, the room acoustics, the speakers accent and speaking style, the speaking domain (i.e. the contents of the recordings) etc.

3. Iterative MAUS

3.1. Method

The here proposed extension `maus.iter` tries to adapt the acoustical HMM of `maus` iteratively during the segmentation process to a defined set of recordings (called *target material* in the following). In a nutshell `maus.iter` does the following:

1. Initialization: Take a speaker independent HMM set for the target language and make a temporary copy of it (for instance, for German use the build-in standard HMM set).
2. Run `maus` over all data available of the target material using the temporary HMM set and store the resulting S&Ls.
3. Compare the S&Ls to the previous iteration; if there are no changes or the number of iterations exceeds *maxiter*, terminate and output the current S&Ls.
4. Otherwise extract the label sequences from the S&Ls.
5. Run *one* iteration of a Viterbi re-estimation based on the temporary HMM and the extracted labelling sequences over the target material.
   We use a complete re-estimation (*segmental-k-means*) to lower the computational effort. The effect is the same as using the S&L and running a Viterbi training on each phoneme class individually, because `maus` uses Viterbi for the alignment just as the Viterbi re-estimation and therefore calculates exactly the same state occupancies.
6. If the number of instances for a phoneme class is higher than a constant threshold *minsegments*, replace the parameters *mean, variance and transition probabilities* in the temporary HMM with the new estimated parameters.
7. Go to 2.

It is obvious that the selection of the target material as well as the constant *minsegments* is crucial for the outcome of this procedure.

3.2. Selection of Target Material

A target database might contain very homogeneous or inhomogeneous recordings. Depending on that a division of the database into appropriate target materials might improve the overall MAUS performance.

The obvious selection of target material from the viewpoint of the acoustical modeling would be the data of one speaker. However, this implies that the user of `maus.iter` has access to a sufficient number of recordings of a target speaker which is not often the case (the question of how much is sufficient will be discussed shortly). However, there are other possible selections of the target material:

- recordings from a certain acoustical condition; for instance if the target database consists of multichannel recordings with three different microphones/channels/codings, it might be possible to divide the material into three target materials and apply `maus.iter` separately on each of the groups.
- recordings of groups of dialect speakers; for instance if the speakers of the target database are labelled into 9 different dialect classes, it might be possible to chose 9 target materials with mixed speakers who all show the same dialectal variety.
- recordings from a certain time period; for instance a target database might contain recordings from 1980 and from 2000.

A more detailed separation of the target database with regard to the acoustical properties will help the `maus.iter` method. However, there is always the trade-back of getting smaller and smaller recording sets for each run of `maus.iter`. So, there will always be an optimal compromise between a more fine grain division and the size of the smallest target materials. The point of this compromise depends on the overall size and the internal structure of the target database and must be determined by trial-and-error. However, there are some basic constraints regarding the minimal size of a target material that might help in the decision.

3.3. Required Instances per Phoneme

The constant parameter *minsegments* determines which phone class models will be re-trained by `maus.iter`. If the phone count within the initial `maus` segmentation for a phone class falls below this threshold, the model will remain untouched. The problem here is that we cannot give an universal value that will be true for all target materials. The value will depend on the structure of the used HMM (i.e. the number of parameters), on the distribution of phone classes within the target material and, of course, on the intrinsic variability of speech features within the phone class. Also, it might be better to choose a *minimum state occupancy* instead of *minsegments* and determine for each state of each phone HMM, whether to re-train the contained emission probability function or not.

Although we cannot give a universal solution to this problem, we conducted some experiments on our benchmark database which are presented in section 5. which give some hints about how to treat this problem.

3.4. Computational Effort

The computational effort of `maus.iter` is roughly that of `maus` times the maximum number of iterations.

For example, to segment a target material of 3360sec length with 34630 phone segments with 20 iterations

`maus.iter` runs for approx. 3 hours on a 900MHz Linux platform.

4. German Benchmark

To evaluate the performance of `maus` and `maus.iter` we use a subsection of the German Verbmobil I corpus (Burger et al., 2000),(Weilhammer et al., 2002). The total benchmark contains 401 turns of face-to-face dialogue recordings with 8232 words of 22 speakers and a total length of 56min. The benchmark is further divided into a development set (DEV) with 4 speakers and a test set (TEST) containing the remaining speakers. The TEST set was manually segmented and labelled by a large group of trained phoneticians, while the complete DEV set was manually segmented and labelled four times by four independent labellers. All data are taken from the Verbmobil volumes VM2, VM7 and VMBONUS which may be ordered by BAS for reference experiments[1].

To compare two different S&Ls we use the *symmetric accuracy* (SA) as proposed by (Kipp, 1999), p. 128 which is basically the mean value of the two possible asymmetric accuracies (AA) where first the one S&L and then the other S&L are taken as the reference.

$$AA = \frac{N_{ref} - N_{rep} - N_{ins} - N_{del}}{N_{ref}} \quad (1)$$

$$SA = \frac{AA_{ref=1} + AA_{ref=2}}{2} \quad (2)$$

To calculate the relative performance of MAUS results we usually determine the inter-labeller agreement of human labellers on the DEV set and then normalize the SA of MAUS on the TEST set to yield a percentage on how well MAUS performs compared to humans. However, to simplify the the presentation of the following results with regards to the statistical significance we will only present the SA values in this paper.

5. Benchmark Results

5.1. Iterative S&L of DEV and TEST set

In this preliminary experiment we did not select any specific target materials from the benchmark data but simply ran `maus.iter` over the combined DEV+TEST set and calculated the combined SA on DEV, TEST and DEV+TEST set. The constant parameter *minsegments* was set to 20 in this experiment; this caused three phonemic classes to be excluded from the re-training: /e/ and /Z/ (SAM-PA) and the garbage model used for all kinds of background noise events.

As can be seen in table 1 `maus.iter` outperforms the traditional `maus` method but the differences are not really significant.

5.2. Iterative S&L of Speaker Sets

We repeated the experiment two times by selecting only the recordings of two speakers (HAR,REA) of the TEST set. Note that the SA values are now determined exclusively from the speaker subsets which are considerably smaller than in the previous experiment.

[1] www.bas.uni-muenchen.de/Bas

SA	DEV	TEST	DEV+TEST
maus	79.81%	78.55%	79.25%
maus.iter	80.69%	79.03%	79.96%
segments	7205	27425	34630
sig-level	0.100	0.100	0.050

Table 1: Symmetric accuracy on un-divided data sets; *minsegments*=20.

SA	HAR (male)	REA (female)
maus	79.80%	80.07%
maus.iter	81.34%	80.86%
segments	2394	2853
sig-level	0.100	-

Table 2: Symmetric accuracy on target materials of speakers HAR and REA; *minsegments*=20.

Again there seems to be a positive tendency but we cannot proof a significant improvement mainly because the number of segments in the sets is too small.

5.3. Data Dependency

We calculated the SA for the speakers HAR and REA over a range of roughly twenty subsets of increasing size. Figure 1 shows the SA in relation to the number of segments used in `maus.iter`. Note that the SA is always calculated over the total target material of the speaker to get comparable values. Both speakers show a minimum of SA at approx. 500 segments. Then the SA increases again and seems to converge for speaker HAR but not for speaker REA. A possible explanation could be that the threshold *minsegments* is to high for a successful re-training of the phone models of speaker REA. To verify that the shown effect is not an arbitrary result caused by the sequence of segments in the target materials we repeated the experiments several times with randomized sequences of the same tar-

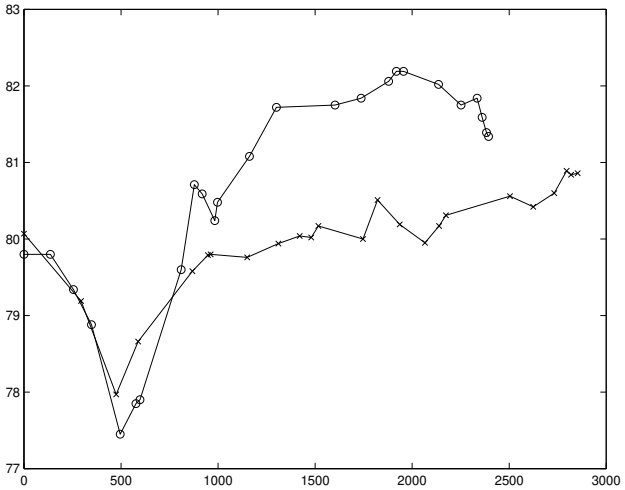

Figure 1: SA values for speakers HAR (o) and REA (x) over increasing number of segments; *minsegments*=20.

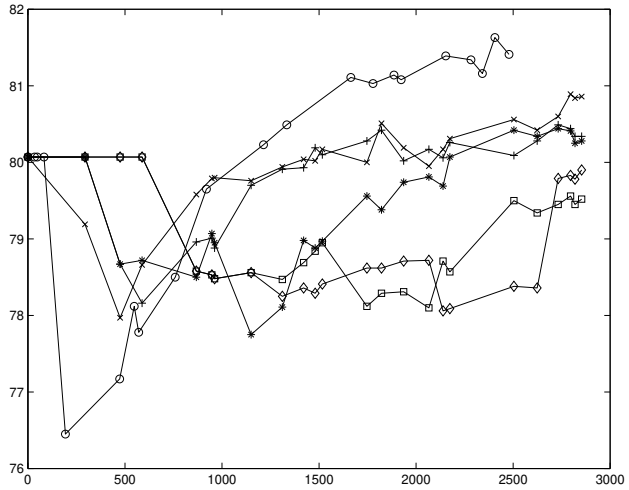

Figure 2: SA values for speaker REA with *minsegments* set to 10 (o), 20 (x), 30 (+), 50 (*), 75 (square) and 100 (diamond).

get materials. We always found the prominent dip of SA at about 500 segments and the same converging target values later on.

5.4. Minimal Number of Segments

We then varied the threshold *minsegments* to see the influence on the development of the SA. Figure 2 shows the SA values of the speaker HAR for $minsegments = 10, 20, 30, 50, 75, 100$. As can be seen the best improvement (significance level 0.1) is reached with *minsegments*=10. However, in an analog experiment using the data of speaker HAR we found the optimal improvement at *minsegments*=20.

6. Discussion

The iterative MAUS method seems to outperform the traditional algorithm under certain constraints. The main factor seems to be the amount of data available for the target database. As can be seen in the results of the previous section our German benchmark data are too small to achieve significant improvements. As mentioned earlier we found very encouraging results using large samples from Australian English, but unfortunately we don't have any manually controlled benchmark data within this data set and therefore cannot present any quantitative results about these improvements. We estimate the minimum of speaker data necessary to achieve considerable improvements to 20 min (data of benchmark speakers HAR and REA was 4 and 6 min).

Another critical factor is the threshold *minsegments*. There seems to be no 'global' threshold that holds true for all speakers. Also this threshold is very likely dependent on the structure of the used HMM in the MAUS system. We recommend to set *minsegments* in the range of $10 - 20$.

7. MAUS Download Package

Since Oct 2003 MAUS is a public domain software that may be used for any scientific or educational purposes. The `maus` package can be download from:

ftp://ftp.bas.uni-muenchen.de/pub/BAS/SOFTW/MAUS

The package is suitable for all Linux variants and contains all necessary data, scripts and documentation to use *maus* or *maus.iter* on German speech data. The installation requires the *HTK toolkit*[2] and *sox*[3] to be installed on your platform.

The *maus* script will read a canonical pronunciation in German SAM-PA from command line or from a BAS Partitur File (BPF) and calculate a MAU tier or a Praat TextGrid file with the resulting S&L. *maus* will accept signal files in *NIST SPHERE* or *WAV* format with different sampling rates (signals will be re-sampled using *sox polyphase* to 16kHz sampling rate before segmentation). The package contains a set of German HMM trained on the Kiel Corpus and two different rule sets: statistical rules trained on Verbmobil; phonological rules without training.

Any bug reports, hints, improvements and success stories provided by users of `maus` are very much appreciated (bas@bas.uni-muenchen.de).

8. Acknowledgments

Parts of this research has been supported by the German Federal Ministry of Education and Research, grant no. 01IVB01 (BITS).

9. References

N. Beringer. 2002. Regeladaptive kategoriale Analyse von Spontansprache. *Dissertation, Ludwig-Maximilians-Universität München*.

N. Beringer. 2003. Independent Automatic Segmentation by Self-learning Categorial Pronunciation Rules. *in: Proceedings of EUROSPEECH 2003, Geneve*.

S. Burger, K. Weilhammer, F. Schiel, H.G. Tillmann. 2000. Verbmobil Data Collection and Annotation. *in: Verbmobil: Foundations of Speech-to-Speech Translation*, edt. W. Wahlster, Springer Berlin New York, page 537-549.

J.M. Kessens, H. Strik. 2001. Lower WERs do not guarantee better transcriptions. *in: Proceedings of the Eurospeech 2001*, pp. 1721-1724.

A. Kipp. 1999. Automatische Segmentierung und Etikettierung von Spontansprache. *Dissertation, Ludwig-Maximilians-Universität München*, Shaker Aachen.

A. Kipp, M.-B. Wesenick, F. Schiel. 1996. Automatic Detection and Segmentation of Pronunciation Variants in German Speech Corpora. *in: Proceedings of the ICSLP 1996, Philadelphia*, page 106-109.

F. Schiel. 1999. Automatic Phonetic Transcription of Non-Prompted Speech. *in: Proceedings of the ICPhS 1999, San Francisco*, page 607-610.

K. Weilhammer, F. Schiel, and U. Reichel. 2002. Multi-tier annotations in the Verbmobil corpus. *in: Proceedings of the 3rd Int. Conf. on Language Resources and Evaluation, Las Palmas, Spain*, page 912-917.

St. Young et al. 1995. The HTK Book. *Cambridge University Press*.

[2] http://htk.eng.cam.ac.uk/

[3] http://www.spies.com/Sox/